Real-time Navigation Recommendations:
Integrating N-gram based Pattern Extraction and Site Content

Diamanto Oikonopoulou

National Technical University
of Athens (NTUA)
Telamonos 7-9 str.
GR-176 71, Kallithea, Athens
mikonomopoulou@telepassport.gr

Maria Rigou

Computer Technology Institute
Riga Feraiou 61 str.
GR-262 21, Patras
&
Computer Engineering and Informatics Department
University of Patras
GR-265 04, Patras
rigou@cti.gr

Spiros Sirmakessis

Computer Technology Institute
Riga Feraiou 61 str.
GR-262 21, Patras
&
Dep. of Applied Informatics in Administration and Economics
Technological Educational Institution of Messolongi, GR-30200,
Messolongi Greece
syrma@cti.gr

Athanasios Tsakalidis

Computer Technology Institute
Riga Feraiou 61 str.
GR-262 21, Patras
&
Computer Engineering and Informatics Department
University of Patras
GR-265 04, Patras
tsak@cti.gr

Abstract

Understanding and modeling user online behavior, as well as predicting future requests, remains an open challenge for researchers, analysts and marketers. In this paper, we propose an efficient prediction schema based primarily on the extraction of sequential navigation patterns from server log files. Traversed paths are monitored, recorded and cleaned before being completed with cashed page views. After session and episode identification follows the construction of *n*-grams. Prediction is based upon a *5+ n*-gram schema with all lower level *n*-grams participating, a procedure that resembles the construction of an *All 5th*-order Markov Model. The schema resolves the typical dilemma between precision and applicability by deploying the site's content and structuring for outputting a prediction for every possible state with small loss in precision. Moreover, the paper explores the potential extension of the schema to allow for fine-tuning between usage-based and semantic-based predictions by incorporating semantic web technologies.

1 Introduction

In recent years, the exponential growth of the World Wide Web allowed fast and immediate access to large amounts of information, as well as a wide delivery of novel services including e-commerce, e-learning and entertainment (Huberman, Pirolli, Pitkow & Lukose, 1998). This growth, in combination with the large number of its users on a daily basis, has les to corporate exploitation and development upon the Internet. Web-based business activities have gained ground towards the more traditional enterprise models, mostly due to certain newly supported features, such as the capability of personalized interaction, electronic marketing, and innovative CRM techniques. The maturity of web technology has led to the evolution of Internet sites from static data warehouses to dynamic and adaptive information portals. In spite of its substantial advantages, the Internet still suffers from some major, unresolved problems that complicate the aforementioned transactions thus decreasing their quality of service. Among others, unacceptably long retrieval times, significant latencies in data transmission, misplaced hyperlinks, and even mal-structured sites, are all factors that may cause user discomfort, irritation and high abandonment rates. Understanding and modeling user behavior remains an open challenge for researchers, analysts and marketers and providing of effective user guidance taking into account individual goals and preferences, is now considered as a crucial requirement. The related literature offers a variety of strategies for addressing this requirement. Many of these strategies are based on the off-line semantic analysis of recorded user requests with the purpose of identifying navigational patterns of users traversing the fundamental structural dimension of the web (in terms of successive page visits through interconnecting links). In the long run, this process aims at the proper restructuring of web sites that will assure improved functionality, based on the extracted knowledge from the trails web users leave behind them and the probabilistic estimations of prospective accesses.

Web usage mining has a central role in this area. The term addresses the application of data mining techniques for detecting correlations among web requests (indicating user objectives) and extracting useful knowledge from web data repositories (Cooley, 2000; Kosala & Blockeel, 2000; Liu, Jiming & Zhang, 2004). Web usage mining application areas include personalization, customization or optimization of web sites, web caching/prefetching and business intelligence. The idea behind all of them is the ability to *predict* upcoming user requests, by observing recorded navigation patterns. For web usage mining, the input data reflect a realistic and unbiased description of user interest(s) and the extraction of usage patterns allows the dynamic construction and updating of user profiles and thereby the computation of real-time page recommendations. It also provides detailed feedback on user behavior (Cooley, 2000), creating a cohesive basis for redesigning decisions, while it can also be used for the application of web prefetching and caching techniques aims at improving network traffic, thus reducing latencies in data transmission and overload on servers (Nanopoulos, Katsaros & Manolopoulos, 2003; Bonchi, Giannoti, Gozzi, Manco, Nanni et al., 2001). More specifically, Nanopoulos et al. (2003) propose a predictive prefetching mechanism, based upon the extraction of proper navigation patterns and Bonci et al. (2001) develop an intelligent web caching architecture, adaptable to the recorded behavior of users and their access patterns.

In this paper, we propose an efficient prediction schema for future web requests, based on the extraction of sequential navigation patterns from server log files, combined with structural and semantic information on the content of web site pages (thematic categorization). Our goal is to achieve request predictions with competitive precision levels, while maintaining high applicability. The proposed schema resolves the typical dilemma between precision and applicability by resorting to site content and structuring for providing a prediction for every possible state, even when usage data cannot provide a suggestion, with small loss in precision. The paper is organized as follows: section 2 provides a brief overview of the related work and references to the techniques used in our approach. The proposed prediction schema is described in section 3 and section 4 contains a case study with experimental results. Section 5 suggests an extended version of the schema that incorporates semantic web technologies and section 6 concludes the paper.

2 Prediction based on Web Usage Mining

Research has indicated that there exist strong statistical regularities among the surfing patterns of web users (Huberman et al., 1998; Liu et al., 2004; Davison, 2002). The purpose of prediction based on web usage mining is to identify proper mechanisms that take advantage of the large volume of data users leave behind while navigating the web and use them for reaching prediction decisions.

Prediction models may be approached from a *data mining* or a *distributed systems* perspective (Li, 2001). In the first case, prediction models are further distinguished based on the pattern recognition techniques they use as either prediction based on clustering, prediction based on classification, prediction based on frequent itemsets and association rules, or prediction based on sequential pattern discovery. More specifically:

- *Clustering* is used to group together items that have similar characteristics. In the context of Web mining, we can distinguish two categories of items: users and pages. Page clustering identifies groups of pages that seem to be conceptually related according to the users’ perception. User clustering results in groups of users that seem to behave similarly when navigating through a Web site. Such knowledge is used for e-commerce for market segmentation purposes and more generally for personalization.
- *Classification* is used for associating objects with one of the classes in a predetermined set (of classes). It is a two-phase procedure that has to go through a training step with a set of objects assigned to the given set of classes, before concluding to a set of discriminating attributes that will be used in the next phase to automatically determine class membership for a new object. In the web domain, classes usually represent different user profiles and classification is performed using selected features that describe each user in terms of a corresponding profile, or category of common attributes, and thus a common anticipated behaviour (Davison & Hirsh, 1998). The most widely used classification algorithms comprise decision trees, naïve Bayesian classifier, and neural networks.

- *Association rule mining* is a technique for discovering interesting associations or correlations among a large dataset (Li, 2001) and provides co-occurrence based prediction. In the web domain, association rules are used for revealing correlations between pages accessed together during server sessions. Such rules indicate possible relationships between pages that are often viewed together even if they are not directly connected (i.e. linked), or reveal associations between groups of users with similar interests. Yang, Li and Zhang (2001), propose an association-based prediction model for caching and prefetching. In some cases, association rules and classification are used together resulting in identifying *class association rules* (Liu, Hsu & Ma, 1998). The main disadvantage of association rules is that they propose a set of high-relevance probability objects that might have prohibitively large size.
- *Sequential pattern discovery* is an extension of association rules mining in that it reveals patterns of co-occurrence but it also incorporates the notion of time sequence. In the web domain such a pattern might be a web page or a set of pages accessed immediately after another such set.

From the distributed systems perspective, the most dominant techniques deal with sequential pattern recognition through machine-learning. The basis of these techniques is the construction of a predictive model that suggests future events based on past experience (web access patterns). Under this view, prediction differs from the data mining approach in that it is prediction of future actions without concern for interactivity or immediate benefit (Davison, 2002). A problem-solving framework used for web document prediction and retrieval, is Case-based Reasoning (CBR). Yang et al. (2001), propose a server-side CBR application, aiming at improving system performance (i.e. reduction of latencies and network load) through prefetching while Schiaffino and Amandi (2000) combine CBR techniques with Bayesian networks.

In statistical natural language processing, an ordered sequence of *n* items is defined as an *n*-gram. In the area of web usage mining, *n*-grams are correlated with time-ordered sequences of user accesses, and thus represent proper subsets of user sessions. Spiliopoulou, Pohle and Faulstich (1999), generalize the definition of an *n*-gram, proposing the use of a *g*-sequence, i.e. a sequence of events and wildcards. There exist two types of *n*-gram-based prediction models: point-based and path-based. The former rely their prediction decision on the currently observed user action (request), concluding therefore in lower prediction accuracy. Path-based models (Su, Yang, Lu & Zhang, 2000; Frias-Martinez & Karamcheti, 2002) take into account the last *n* recorded accesses to reach a prediction, capturing this way the temporality and the sequence of web accesses. Although the prediction precision of path-based models is much more satisfactory, they might suffer from low applicability, due to the infrequent appearance of long-length patterns. Experimental results have shown that for *n* greater than 4, precision's upper bound does not vary noticeably, while applicability decreases drastically (Su et al., 2000). The same conclusion is supported in (Deshpande & Karypis, 2004), although it is based on a Markovian model rather than *n*-grams.

Markov models (MM) have been used extensively in the field of stochastic processes (Papoulis, 1991), and are quite capable of tracking the likelihood of varying *n*-grams, in a state space encoding. Variations of MMs comprise Time MMs, sharing the assumption that the current request depends only on ita precedent request, K^{th}-order MMs, that rely their prediction on the last *n* observed requests, Space MM's that add structural constraints in Time MMs and last but not least, Linked Space MMs, which combine structural and auxiliary data (Albrecht, Zukerman & Nicholson, 1999; Deshpande & Karypis, 2004). The disadvantages of higher-order Markov Models summarize in their non-negligible complexity, their requirements in terms of storage, their insufficient coverage and occasionally, their worse predictive accuracy, compared to lower MM's (Davison, 2002; Deshpande & Karypis, 2004). As an alternative, a hybrid *All* K^{th}-order MM may be used that combines different order MMs in a way that the resulting model has a low state complexity, improved prediction accuracy, and retains the coverage levels of its components (Deshpande & Karypis, 2004). Davison (2002) has proposed a prediction algorithm for upcoming user commands, based on the simple Markov assumption, e.g. Time MM, as well as an effective mechanism for representation and storage of higher order MM's, through a tree structure called trie. In (Jacobs & Blockeel, 2002) a Time MM is also used for prediction and the extraction of longer decision rules in the cases with accurate prediction.

3 The Proposed Approach to Prediction

The prediction schema proposed in this paper -an earlier version of this work can be found in (Oikonomopolou, Rigou, Sirmakessis & Tsakalidis, 2004)- is based on the extraction of sequential navigation patterns from recorded server log data, while integrating structural and semantic information on the content of web site pages (thematic categorization). Our goal is to encapsulate the internal motivations and ultimate objectives of varying user profiles, into corresponding access patterns, so as to allow for reliable predictions. The mining process (i.e. the extraction of semantically meaningful patterns) uses as input page access requests, sent from anonymous, non authenticated users, for whom the system has no information regarding individual goals or characteristics. This applies in many cases to user accesses to commercial sites and portals that choose not to jeopardize user privacy or drive customers away by requiring registration and login procedures. A schematic representation of the complete process can be found in Figure 1.

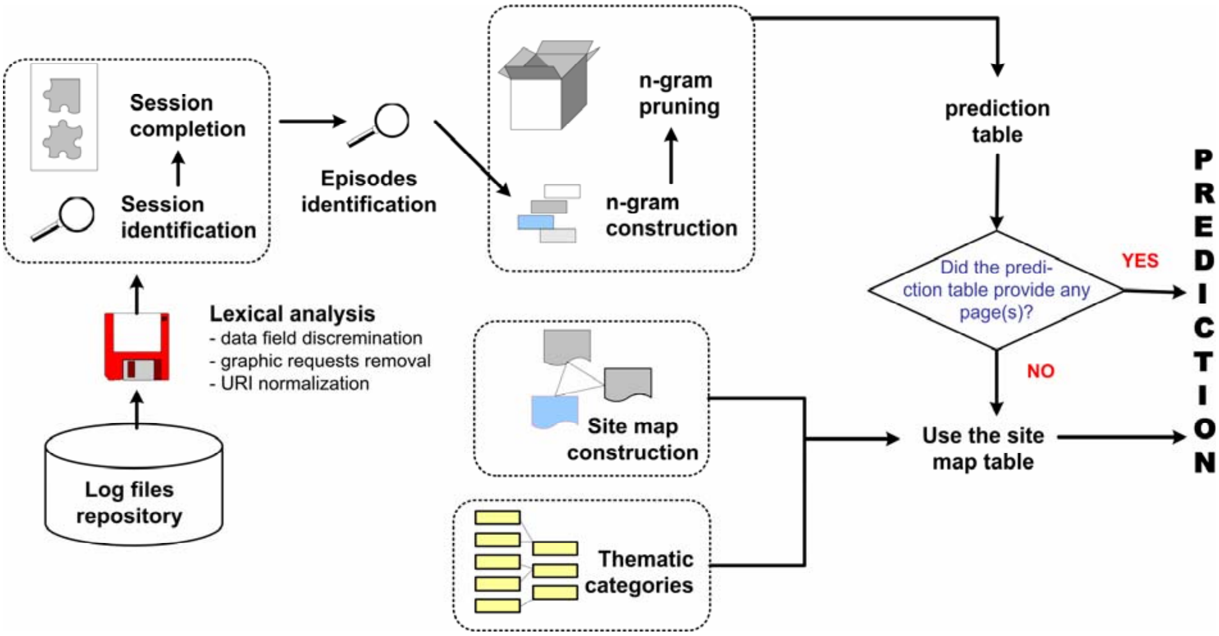

Figure 1: The overall prediction process

The process of pattern extraction (or web mining process) executes in two phases. The first phase includes data cleaning and log files parsing into proper data fields. Data cleaning regards stripping out of the dataset requests for graphics files, as well as page misses (i.e. requests with error status code). Before proceeding to the next phase, all URI fields of the remaining page requests are normalized. The second phase consists of four successive steps: (1) user session identification and extraction from the dataset, (2) session completion with cached page views, (3) episode extraction from user sessions and (4) *n*-grams construction. The term session refers to "*a delimited, time-ordered set of user clicks*" (Lavoie & Nielsen, 1999). Session identification is typically based on either time or structure related criteria. Berendt et al. (2001) have introduced a set of heuristics for session identification using a threshold on the overall click-stream duration (session time-out) that should not be exceeded, a threshold on the duration of a unique request (page time-out) and a referrer-based criteria, under which a request is assigned to a session, if and only if, its referrer has been previously recorded in the same session, as a request. Our working assumption is that each different agent type for an IP address represents, at least one, discrete session, as in (Cooley, 2000) and (Pirolli, Pitkow & Rai, 1996). In addition, we make use of the referrer based criteria, assuming that the time interval spent on a single page file should not exceed a given threshold (otherwise, a new session has started). Finally, in case of multiple candidate sessions, the assignment of requests with the same IP address and agent to one of them is determined by measuring the distance between the request and each session. The *distance* between a request *r* and a session *S*, is defined as the number of links needed to be traversed from the last recorded page view of *S*, in order to obtain the referrer field of *r* as a request in the same session (Cooley, 2000). The request is assigned to the session of minimum such distance. The algorithm for session identification is presented in Table 1.

Table 1: Session identification

```
Let $H_i = \{f_1, f_2, ... , f_n\}$ denote a time-ordered session history
Let $<l_j, f_j, r_j, t_j>$ denote log entry, request, referrer and time (at which the request was received) respectively
Let $T$ denote a session timeout

Sort data by IP address, agent and time
  for each unique IP/agent combination do
    for each $l_j$ do
      if $((t_j - t_{j-1}) > T)$ or $r_j \notin \{H_0, H_1, ... , H_m\}$ then
        increment $i$, add $l_j$ to $H_i$
      else
        assign $l_j$ to $H$, such that distance $(H, l_j)$ is min
```

The extracted sessions must be complemented with cached page views that were actually traversed, although not recorded in the log files (see algorithm presented in Table 2). This is caused by user backtracking during navigation and is detected by examining whether the referrer field of a page view differs from the previously observed request. Cooley (2000) tracks cashed page accesses by keeping a certain number of recent page requests in a stack and in the case of a stack miss, resorts to a full history search. In this work we decided to perform full history search and skip the intermediate stack access, since experimental results showed that the full history search does not cause noticeable performance reduction, as sessions are not expected to be long.

Table 2: Session completion heuristic

```
Let $S$ denote a session of page views $\{V_1, V_2, ... , V_n\}$
Let $r_u$ denote the referring page file of $V_i$
Let $stk$ denote a stack
Let $S_{temp}$ denote a temp session

for each $V_i$ $(i = 2,...,n) \in S$ do
   if $r_u$ != $V_{i-1}$ then
     push $\{V_1, V_2, ... , V_{i-1}\}$ into $stk$
     set $not\_found$ = true
   if $not\_found$ = true then
     while (($u_s$ = ($stk$. pop)) != null and $not\_found$ )
       add $u_s$ into $S_{temp}$
       if $u_s = r_u$ then
          set $not\_found$ = false
    if $not\_found$ = false then
      add $S_{temp}$ to $S$ in $i$-1
```

Session completion is followed by episode identification. An episode is defined as *"a subset of related user clicks within a user session"* (Lavoie & Nielsen, 1999). Although this task is optional and it is often unconsidered in practice, episodes should be regarded as navigational subsets of significant semantic value that depict a well-formed snapshot of user's orientation and desire. A *general episode model* identifying episodes in sessions, based upon page type discrimination (auxiliary or media page), is proposed in (Cooley, 2000) and concludes to media-only or auxiliary-media episodes. Another alternative is to consider episodes as strict subsets of media pages, in order to encompass different pages correlations. In this work we adapted the *maximal forward reference* method -as described in (Chen, Fowler & Fu, 2003)- whereat an episode is defined as a time-ordered click sequence up to the request before a backward reference occurs. This is justified by the assumption that forward references may be used as reliable indications of what the user is looking for, while backtracking can be considered as ‘noise’ to the semantic interpretation of user navigation.

The second phase of data processing proceeds with the formation of *n*-grams from the extracted episodes. We specifically focus on 3rd to 5th-order *n*-grams and their suffices. These lengths were chosen as an equilibrium factor between precision and applicability, since long *n-grams* tend to increase prediction accuracy with a considerable loss in applicability. This is also backed up by the observation that in most real-world implementation scenarios, long traversal sequences repeat less frequently than shorter ones. 1st-order *n*-grams are also included in order to sustain high scores in coverage. The proposed algorithm generates an index table *T* that lists all distinct *n*-gram couples and their subsequent request, as observed in the dataset. Each record in the index table includes an *n*-gram, its suffices and a corresponding support value (indicating the occurrence frequency of the specific *n*-gram). An *n*-gram consists of page identifiers separated by a proper delimiter. After extracting all possible *n*-grams, we compute the overall mean support in order to prune out of the index table all rows with support below average. *N*-grams whose the lower-order proper subsets present a noticeably higher support value (but still over the threshold), are also removed. As a consequence, a matching (*n*-1)-gram will be preferred over a matching *n*-gram, in case it demonstrates better support. Even though in most cases, higher order *n*-grams perform better in terms of precision, the prediction is based on a lower order *n*-gram, if the prerequisite (significantly higher support) stands. Obviously, 1-grams are not affected by this last step, which is legitimate so that coverage maintains a satisfactory level. No *n*-grams are pruned out of the dataset due to their unary suffix, mostly because 1-grams rely their decision on a single observed request.

The remaining set of *n*-grams provides the final patterns for obtaining prediction(s), concluding to a prediction table *P*. Deshpande and Karypis (2004) propose various methods for this step, based on namely *support-pruned*, *confidence-pruned* or *error-pruned* models. The support-pruned method performs pruning under a minimum support criteria. The *confidence-pruned* method takes also into account the probability distribution of actions outgoing an *n*-gram, using statistical techniques, and the error-pruned model uses a validation step to estimate the error associated with each *n*-gram and prunes out of the set those *n*-grams whose proper subsets perform better (at lower error rates). We have adapted the support-pruned method with the choice of a support threshold relying on the assumption that decision rules with low occurrence frequency in the training set, are expected to demonstrate a low prognostic performance, as well.

The prediction phase following, is based upon a 5+ *n*-gram schema with all lower order *n*-grams (of length 5, 4, 3, 2 & 1) participating in it. This procedure resembles the construction of an *All 5th-order Markov Model*. Generally, higher order *n*-grams receive higher priority by the prediction algorithm. In case that an observed sequence cannot be matched with some recorded *n*-gram, the prediction algorithm searches for matching (*n*-1)-grams, taking into account only the *n-1* last requests of the sequence. When attempts for matching *n*-grams (n>1) are unsuccessful, the algorithm searches the prediction table *P* for the matching 1-gram (corresponding to the last recorded page file in the sequence).

If there exists no matching *n*-gram in table *P* (n=1,2,3,4,5), the algorithm proceeds with examining the link structure of the web site and constructing a site-map. The site-map has the form of a connected graph that represents the site’s internal hyperlink structure, where page files are inter-connected through links and there exists a starter node that refers to the site’s home page. Moreover, the algorithm uses the semantic information that has been manually associated with the page files, by assigning them to the proper thematic category from a set of predefined ones. Given a sequence *seq* that cannot be matched with any *n*-gram in *P* and the last observed request from *seq* being *l*, the prediction algorithm searches for all *l*’s outgoing links stored pair-wise in the *site-map,* in the form of *(l, p)*, and outputs as prediction the page with the highest support value in the training dataset, that also belongs in *l*’s category. The rationale is based on the fact that in absence of a suitable matching pattern, we should search upon all potential single step transitions of *l* -according to the site-map - and choose the most frequently observed, assuming that the user will keep on navigating through pages of the same thematic category. This way, we are able to produce predictions even for cases that there are no recorded user patterns in the training dataset, concluding to an overall full coverage scheme.

4 Experimental Results

With the purpose of testing the proposed prediction schema, we performed a case study by considering single action prediction, i.e. prediction of the page file that will be requested immediately after a recorded sequence of requests in order to recommend it. This approach is quite accommodating for both evaluating prediction schemas with varying lengths and reaching comparative conclusions. The case study used log data collected by monitoring users with different profiles, objectives and needs, navigating in a multi-topic electronic magazine. The e-magazine was composed of 143 pages, categorized in a 4-level hierarchy of thematic topics. An arbitrary set of the log data was used as a training set and the remaining as test data and the (single action) prediction schema was based on an *all 3rd, 4th and 5th order n-gram model*. As already mentioned, an *all nth*-order *n*-gram model makes a prediction using an *n*-gram and all its proper subsets. In cases where the recorded sequence can not be supported by an existing *n*-gram, we base the decision on the site-map table and the page categorization, as described in the previous section. It is obvious that an *all 5th order model* includes all lower order models.

Comparisons in the performance of different order models were based on the *precision* (or *accuracy*) and *applicability* values of each schema. Precision is defined as the number of correct predictions P^{+} divided by the number of feasible predictions $P^{+} + P^{-}$ (where P^{-} refers to the number of unsuccessful predictions), that is:

$$precision = \frac{P^{+}}{P^{+} + P^{-}} \qquad (1)$$

Applicability is defined as the number of feasible predictions divided by all cases R that are used as input to the prediction process, that is:

$$applicability = \frac{P^{+} + P^{-}}{R} \qquad (2)$$

Note that a case in R that the algorithm failed to produce a prediction for is not assigned to neither P^{+} nor P^{-}. Figure 2 plots precision and applicability for the different *n*-gram models under the hypothesis that the site-map and the thematic categorization were not used in the prediction process (pure model). Figure 3 presents the resulting values when the site-map and the content semantics were taken into account. As depicted in Figure 3, in the latter case prediction achieves full coverage.

In Figures 2 and 3, the *all 3rd* and *all 4th order* models have been grouped together as they produce the same precision and applicability values. This is justified by the fact that in the majority of cases, the pruning step resulted in removing from the data set the same *n*-grams for both *all 3rd* and *all 4th order* models. This in turn, leads to the calculation of similar prediction tables *P* and therefore precision values with minor deviations.

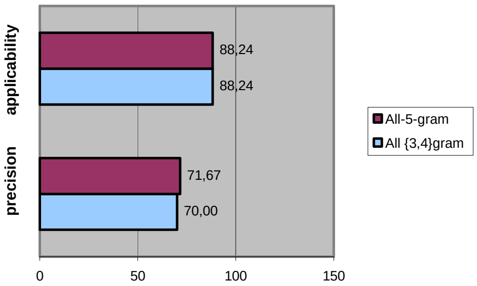

Figure 2: Precision and applicability for the pure *all nth-order n*-gram model

Precision upper bounds for the *all 5th-order model* appear greater than these of lower order models, due to the fact that the former rely their prediction decisions upon the *n*-grams of the latter, alongside with a number of additional ones. This observation does not conflict with our assumption that high order *n*-grams, tend to demonstrate lower accuracy, since in our schema higher order models incorporate the corresponding lower ones. Thus, if for instance, we base our prediction on 3-grams, 4-grams and 5-grams only, precision drops by approximately 20%, which is a quite drastic reduction.

Concerning the fact that long sequences repeat rarely in the dataset, in most cases prediction is based upon *n*-grams with efficient length. Another significant observation is the decrease of precision and the increase of applicability in the hybrid approach (Figure 3). This fall is justified by the requirement for full coverage. More specifically, applicability increases as its numerator $P^+ + P^-$ increases (Equation 2), which also results in a simultaneous decrease in precision (Equation 1). Finally, it is worth mentioning that the extra overhead imposed by taking the *all 5th-order model* approach is mostly summarized in additional space requirements.

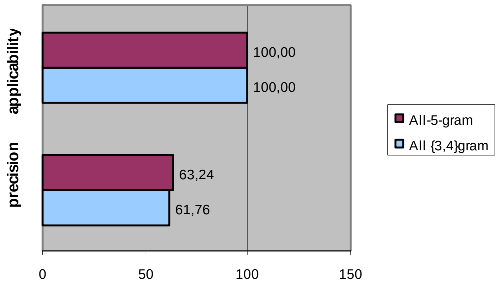

Figure 3: Precision and applicability for the hybrid *all n^{th}-order n*-gram model

The proposed schema demonstrates linear retrieval times -as far as the prediction decisions are concerned- while keeping space and memory requirements low through proper data tabulation. The prediction table is easily updated as the training dataset increases, allowing constant learning on the part of the algorithm. Besides, the schema is flexible enough to support prediction for more than one action with minor tuning. Furthermore, in cases where the schema is used for producing recommendations or applying web prefetching, the fact that it assures full-coverage is a significant asset. Precision's upper bound reached 71,67%, a quite competitive percentage when compared to other prediction techniques. For example, Su et al. (2000) also use an *all 3*-gram model, achieving best case prediction accuracy around 63%, while applicability drops by 40%. Similar results stand for (Frias-Martinez & Karamcheti, 2002), where authors propose a sequential behaviour model for prediction. Deshpande and Karypis (2004) follow an approach similar to ours using a 5th-order Markov-based prediction model that demonstrates accuracy around 50% when tested on log files coming from e-commerce sites. Davison and Hirsh (1998) propose a machine learning algorithm that accomplishes 40% precision in predicting future requests. Yang et al. (2001) propose a CBR technique for web object prediction that offers a prediction accuracy upper bounded by 40%.

5 Fine-tuning between usage data and content semantics

The manual assignment of page files to a set of thematic categories may prove in certain cases time-consuming or even ineffective, especially when applied to dynamic web sites with complex content-to-structure correlations. Moreover, in most real-life cases simple *"belongs_to"* relations do not suffice for representing neither page contents, nor page-to-page similarities. In the experimental setting we have discussed so far, pages represent articles and an article entity should not be described only by its thematic category (and the corresponding sub categories when a hierarchy is used) but also by its author, the date it was written, and so on. In addition, an article may belong to more than one categories, or be related with different categories at different degrees.

All the aforementioned factors call for a more sophisticated representation that will allow the assignment of content semantics to article pages. This can be accomplished by the creation of an ontology that formally describes both the article structural and content attributes. Pages may then be associated with the ontology constructs, thus allowing for computing semantic similarities among them. For this purpose, class instances extracted from the ontology will have to be converted into a vector representation, where each article page is associated with the respective values for each attribute dimension, as in (Mobasher, Jin & Zhou, 2004). This step may require normalization and segmentation of continuous attribute types (such as dates or prices). Thus, for example, in the resulting semantic attribute matrix, each category is a column and each article is assigned weighted values depending on its degree of relevance to the corresponding thematic (sub)category. Prediction decisions on the upcoming article page requests or recommendations may be based on usage similarities –i.e. *UsageSim*(p_i, p_j)-, which is calculated from the prediction table P (as already described), or on (static) semantic similarities –i.e. *SemSim*(p_i, p_j)- between the current article and the rest of the articles that are not found on the current user path. A more flexible approach is to fine-tune between these two similarity measures by selecting an appropriate value for α and calculating the combined similarity of articles p_i and p_j:

$$PageSim\left(p_i, p_j\right) = a * UsageSim\left(p_i, p_j\right) + (1 - a) * SemSim\left(p_i, p_j\right) \qquad (3)$$

When α=0, predictions are solely based on usage similarity, while when α=1 we compute page similarities (and make prediction decisions) based exclusively on their content semantics. Given that the user we examine is currently viewing article page p_i and using again the scenario of single action prediction, the algorithm outputs article page p_j that maximizes *PageSim*(p_i, p_j).

6 Conclusions and future work

In this work we have presented an efficient schema for predicting future web requests on a given web site, based on the extraction of sequential navigation patterns from already recorded log data, combined with the site's internal link structure and the assignment of pages to a set of predefined thematic categories, for providing full coverage prediction (i.e. a prediction for every possible state), while maintaining high accuracy. The schema may be extended to allow for fine-tuning between usage-based and semantic-based prediction by incorporating an ontology infrastructure and assigning semantic metadata and content correlations to page files as described in section 5. Even without incorporating such semantic web technologies, the prediction schema as proposed in section 3, may be further improved. The categorization method used in section 3 treats page files similarly, regardless of whether they are media or auxiliary ones. Auxiliary pages though, present greater support than media ones. This, results in having the algorithm assign higher priority to an auxiliary page (placed on the path that leads to many media pages) when applying site-map based prediction, and thus undermining the related media pages. And though the performance of the hybrid model is lower than the corresponding pure model, in case that a more complex page categorization is used, precision's upper bound is expected to increase. Another refinement can be achieved during episode extraction. Chen et al. (2003) propose linear complexity algorithms for the extraction of generalized episodes that can be incorporated in the proposed algorithm for improved performance.

7 References

Albrecht, D., Zukerman, I., & Nicholson, A. (1999). Pre-sending Documents on the WWW : A Comparative Study. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, 1274-1279.

Berendt, B., Mobasher, B., Spiliopoulou, M., Wiltswire, J., Dai, H., Luo, T., & Nakagawa, M. (2001). Measuring the Accuracy of Sessionizers for Web Usage Analysis. In *Proceedings of the Web Mining Workshop at the First SIAM International Conference on Data Mining*, Chicago, IL, 7-14.

Bonchi, F., Giannoti, F., Gozzi, C., Manco, G., Nanni, M., Pdreschi, D., Renso, C., & Ruggirei, S. (2001). Web Log Data Warehousing and Mining Intelligent Web Caching. *Journal Data and Knowledge Engineering*, 39(2), Elsevier Science B.V., 165-189.

Chen, Z., Fowler, R., & Fu, A.W. (2003). Linear Time Algorithms for Finding Maximal Forward Reference. In *Proceedings of the 2003 IEEE International Conference on Info Tech: Coding and Computing (ITCC03*), Las Vegas, Nevada, 160-164.

Cooley, R.W. (2000). Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. *PhD Thesis submitted to Faculty of the Graduate School of the University of Minnesota*.

Davison, B. D. (2002). The Design and Evaluation of Web Prefetching and Caching Techniques. *PhD thesis submitted to the Graduate School of New Brunswick Rutgers in the state University of New Jersey*.

Davison, B. D., & Hirsh, H. (1998). Predicting Sequences of User Actions. Presented at the AAAI-98/ICML'98 Workshop on Predicting the Future: AI Approaches to Time Series Analysis, Madison, WI, and published in *Predicting the Future: AI Approaches to Time Series Problems, Technical Report WS-98-07*, AAAI Press, 5-12.

Deshpande, M., & Karypis, G. (2004). Selective Markov Models for Predicting Web-Page Accesses. *ACM Transactions on Internet Technology (TOIT)*, 4(2), 163-184.

Frias-Martinez, E., & Karamcheti, V. (2002). A prediction model for user access sequences. In *Proceedings of the WEBKDD Workshop: Web Mining for Usage Patterns and User Profiles, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Huberman, B. A., Pirolli, P. L.T. Pitkow, J. E. & Lukose, R. M. (1998, 3 April). Strong regularities in World Wide Web surfing. *Science*, 280(5360), 95-97.

Jacobs, N., & Blockeel, H. (2002). Sequence Prediction with Mixed Order Markov Chains. In *Proceedings of BNAIC'02 Belgian-Dutch Conference on Artificial Intelligence* (Blockeel, H. and Denecker, M., eds.), 147-154.

Kosala, R., & Blockeel, H. (2000). Web Mining Research: A Survey. *ACM SIGKDD Explorations*, 2(1), 1-15.

Lavoie, B., & Nielsen, H. F. (1999). *Web Characterization Terminology & Definitions Sheet*. Retrieved December 10, 2004, from http://www.w3.org/1999/05/WCA-terms.

Li, T. (2001). Web Document Prediction and Presending Using Association Rule Sequential Classifiers. *A Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Simon Fraser University.* Retrieved March 24, 2004, from http://citeseer.ist.psu.edu/li01webdocument.html

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating Classification and Association Rule Mining. In *Proceedings of Knowledge Discovery and Data Mining*, New York, 80-86.

Liu, J., Zhang, S., & Yang, J. (2004). Characterizing Web usage regularities with information foraging agents. *IEEE Transactions on Knowledge and Data Engineering,* 16(5), 566-584.

Nanopoulos, A., Katsaros, D., & Manolopoulos, Y. (2003). A Data Mining Algorithm for Generalized Web Prefetching. *IEEE Transactions on Knowledge and Data Engineering,* 15(5), 1155-1169.

Oikonomopoulou, D., Rigou, M., Sirmakessis, S., & Tsakalidis, A. (2004). Full-Coverage Web Prediction based on Web Usage Mining and Site Topology. *In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI-2004)*, Beijing, China, 716-719.

Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes*. Mc Graw Hill.

Pirolli, P., Pitkow, J., & Rao, R. (1996). Silk from a sow's ear: Extracting Usable Structures from the Web. In *Proceedings of CHI-96*, Vancouver, 118-125.

Schiaffino, S.N., & Amandi, A. (2000). User Profiling with Case-Based Reasoning and Bayesian Networks. In Open Discussion Track Proceedings of the International Joint Conference IBERAMIA-SBIA 2000, Atibaia, Brazil, 12-21.

Spiliopoulou, M., Pohle, C., & Faulstich, L.C. (1999). Improving the Effectiveness of a Web Site with Web Usage Mining. In *Proceedings of WEBKDD' 99*, Springer-Verlag, Lecture Notes in Artificial Intelligence, 1836, 51-56.

Su, Z., Yang, Q., Lu, Y., & Zhang, H. (2000). What next: A prediction System for Web Requests Using N-gram Sequence Models. In *Proceedings of the First International Conference on Web Information Systems and Engineering Conference*, Hong Kong, 200-207.

Yang, Q., Li, I.T.Y., & Zhang, H.H. (2001). Mining High-Quality cases for hypertext prediction and prefetching. In *Proceedings of the 2001 International Conference on Case Based Reasoning, ICCBR-2001*, Vancouver BC, Canada, Springer-Verlag, Lecture Notes in Computer Science, 2080, 744-756.

Mobasher, B., Jin, X., & Zhou, Y. (2004). Semantically Enhanced Collaborative Filtering on the Web. In *Proceedings of the European Web Mining Forum*, Bettina Berendt et al. (ed.), Lecture Notes in Artificial Intelligence, Springer. Retrieved November 6, 2004, from http://maya.cs.depaul.edu/~mobasher/papers/ewmf04.pdf.